**UE18CS202: Data Structures (4:0:0:0:4)**

**Department of Computer Science and Engineering**

**PES UNIVERSITY**

#of Credits:4 #of Hrs: 56

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class #** | **Chapter Title / Reference Literature** | **Topics to be Covered** | **% of Portion Covered** | |
| **% of Syllabus** | **Cumulative %** |
| **Unit 1 :Overview of Data Structures, Lists** | | |  |  |
| 1 | **Unit 1**  **R1: 1.3,3.2,3.4,**  **4.5,5.1,5.2**  **Appendix C** | Overview of course, Programming Practices,  Definition of Data structures  Classification of Data Structures | 21 | 21 |
| 2 | Revision of Structures, union, enumerated data types  Definition of pointer ,pointer types Functions, Parameter passing for functions, passing a structure to function |
| 3 | Recursion  Example of programs ,Concept of stack frames |
| 4 | Revision of Dynamic Memory allocation  1d array allocation,2d array allocation |
| 5 | List Definition  Array based implementation of list with various operations |
| 6 | Linked List Definition,  Linked List operations –create, Insert, delete, traverse, update |
| 7 | Linked List-Position Based operations |
| 8 | Link List operations – concatenate, merge ,reverse |
| 9 | Definition and Implementation Of Doubly Linked List  Concept of Header nodes, Trailer nodes |
| 10 | Doubly linked list Implementation and operations |
| 11 | Circular Linked List implementation |
| 12 | Multi-List and application of Lists+ Revision of Unit1 |
| **Unit 2 :Stacks, Queues** | | |  |  |
| 13 | **Unit 2**  **R1: 3.1,4.1-4.3, 4.6**  **R2: 2.3,4.5(page 232)** | Stack – Definition, Operations ,implementation approaches, applications in brief | 20 | 41 |
| 14 | Stack – Linked List/Array Implementations |
| 15 | Stack – Applications (post fix conversion) |
| 16 | Stack – Applications (expression evaluation) |
| 17 | Stack – parentheses balancing etc. |
| 18 | Queue – Definition and Operations, implementation approaches, applications in brief |
| 19 | Queue –Array implementation |
| 20 | Queue – Linked list based implementation |
| 21 | Circular Queues – Implementation using arrays |
| 22 | Circular Queues – Implementation using Linked List |
| 23 | Double ended Queue+ Revision of Unit2 |
| **Unit 3 :Graphs, Trees, Binary Trees** | | |  |  |
| 24 | **Unit 3**  **R1: 11,9.1** | Graph Definition and concepts related to graphs, Applications | 18 | 59 |
| 25 | Representations of Graphs  Adjacency Matrix Adjacency List |
| 26 | Graph Traversal using BFS |
| 27 | Graph Traversal using DFS |
| 28 | General Tree Representation  Traversals, Applications |
| 29 | Binary Tree :-Definition, terminologies, Representations |
| 30 | Binary Tree Implementation |
| 31 | Binary Tree Traversals, |
| 32 | Binary Tree Recursive operations |
| 33 | Revision of Unit3 |
| **Unit 4 : Binary Search Tree, Heap Tree** | | |  |  |
| 34 | **Unit 4**  **R1:7.9,9.2, 9.3 ,9.4**  **Appendix B5** | Building and evaluating Expression Tree | 23 | 82 |
| 35 | Binary Search Tree Definition, Terminologies, Applications |
| 36 | Constructing a BST |
| 37 | BST operations : insert ,delete |
| 38 | AVL Tree Definition ,concepts applications |
| 39 | Insert operation in AVL |
| 40 | Delete operation in AVL |
| 41 | Threaded Binary Search Tree |
| 42 | Heap Implementation |
| 43 | Heap operations-Insert ,Delete |
| 44 | Heap Trees – Find Min |
| 45 | Priority Queue using Heap |
| 46 |  | Revision of Unit4 |
| **Unit 5: Tries, Hashing** | | |  |  |
| 47 | **Unit 5**  **R1: 10.2,8.6** | Tries : Definition Implementation | 18 | 100 |
| 48 | Tries : Implementation Applications |
| 49 | Tries : Applications |
| 50 | Hash Table ,Hash Functions |
| 51 | Collision Handling – Open Addressing |
| 52 | Collision Handling – Open Addressing |
| 53 | Collision Handling – Chaining |
| 54 | Collision Handling – Chaining |
| 55 | Revision |
| 56 | Course Summary |

**Literature**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Book Type** | **Code** | **Title & Author** | **Edition** | **Publisher** | **Year** |
| Reference | R1 | *“****Data Structures and Program Design in C****”,*  Robert Kruse, C.L.Tondo, Bruce Leung and Shashi Mogalla,  Second Edition, /PHI, 2015 | 2 | Pearson/PHI | 2015 |
| Reference | R2 | “***Data Structures Using C and C++”*,** Tanenbaum, Langsam, Augenstein Pearson/Prentice Hall , 2nd Edition, 2015 | 2 | Pearson/Prentice Hall | 2015 |